

Minimum Fill-in of Sparse Graphs: Kernelization and Approximation*

Fedor V. Fomin¹, Geevarghese Philip², and Yngve Villanger¹

¹ Department of Informatics, University of Bergen, N-5020 Bergen, Norway.
{fomin|yngvev}@ii.uib.no

² The Institute of Mathematical Sciences, C.I.T Campus, Taramani, Chennai
600 113, India. gphilip@imsc.res.in

Abstract

The MINIMUM FILL-IN problem is to decide if a graph can be triangulated by adding at most k edges. The problem has important applications in numerical algebra, in particular in sparse matrix computations. We develop kernelization algorithms for the problem on several classes of sparse graphs. We obtain linear kernels on planar graphs, and kernels of size $\mathcal{O}(k^{3/2})$ in graphs excluding some fixed graph as a minor and in graphs of bounded degeneracy. As a byproduct of our results, we obtain approximation algorithms with approximation ratios $\mathcal{O}(\log k)$ on planar graphs and $\mathcal{O}(\sqrt{k} \log k)$ on H -minor-free graphs. These results significantly improve the previously known kernelization and approximation results for MINIMUM FILL-IN on sparse graphs.

1998 ACM Subject Classification G.2.2 Graph Theory — Graph Algorithms

Keywords and phrases Minimum Fill-In, Approximation, Kernelization, Sparse graphs

Digital Object Identifier 10.4230/LIPIcs.FSTTCS.2011.164

1 Introduction

A graph is *chordal* (or triangulated) if every cycle of length at least four has a chord, i.e. an edge between nonadjacent vertices of the cycle. In the MINIMUM FILL-IN problem (also known as MINIMUM TRIANGULATION and CHORDAL GRAPH COMPLETION) the task is to check if at most k edges can be added to a graph such that the resulting graph is chordal. That is

MINIMUM FILL-IN

Input: A graph $G = (V, E)$ and a non-negative integer k .

Question: Is there $F \subseteq [V]^2$, $|F| \leq k$, such that graph $H = (V, E \cup F)$ is chordal?

This is a classical computational problem motivated by, and named after, a fundamental issue arising in sparse matrix computations. During Gaussian eliminations of large sparse matrices, new non-zero elements — called *fill* — can replace original zeros, thus increasing storage requirements, the time needed for the elimination, and the time needed to solve the system after the elimination. The problem of finding the right elimination ordering

* The research of Fedor V. Fomin was supported by the European Research Council (ERC) grant “Rigorous Theory of Preprocessing”, reference 267959. The research of Yngve Villanger was supported by the Research Council of Norway.



© Fedor V. Fomin, Geevarghese Philip, and Yngve Villanger;
licensed under Creative Commons License NC-ND

31st Int'l Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS 2011).

Editors: Supratik Chakraborty, Amit Kumar; pp. 164–175

Leibniz International Proceedings in Informatics



LIPIcs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

minimizing the amount of fill elements can be expressed as the MINIMUM FILL-IN problem on graphs [21]. Besides sparse matrix computations, applications of MINIMUM FILL-IN can be found in database management, artificial intelligence, and the theory of Bayesian statistics. The survey of Heggenes [15] gives an overview of techniques and applications of minimum and minimal triangulations.

Unfortunately, the problem is notoriously difficult to analyze from the algorithmic perspective. MINIMUM FILL-IN (under the name CHORDAL GRAPH COMPLETION) was one of the 12 open problems presented at the end of the first edition of Garey and Johnson's book [13] and it was proved to be NP-complete by Yannakakis [26]. Due to its importance the problem has been studied intensively, and many heuristics, without performance guarantees, have been developed [18, 21].

Very few approximation and FPT algorithms for MINIMUM FILL-IN are known. Chung and Mumford [8] proved that every planar, and more generally, H -minor-free, n -vertex graph has a fill-in with $\mathcal{O}(n \log n)$ edges, thus yielding an $\mathcal{O}(n \log n)$ -approximation on these classes of graphs. Agrawal et al. [1] gave an algorithm with the approximation ratio $\mathcal{O}(m^{1.25} \log^{3.5} n/k + \sqrt{m} \log^{3.5} n/k^{0.25})$, where m is the number of edges and n the number of vertices in the input graph. For graphs of degree at most d , they obtained a better approximation factor $\mathcal{O}((nd + k)\sqrt{d} \log^4 n/k)$. Natanzon et al. [17] provided another type of approximation algorithms for MINIMUM FILL-IN. For an input graph with a minimum fill-in of size k , their algorithm produces a fill-in of size at most $8k^2$, i.e., within a factor of $8k$ of optimal. For graphs with maximum degree d , they gave another approximation algorithm achieving the ratio $\mathcal{O}(d^{2.5} \log^4(kd))$. Kaplan et al. proved that MINIMUM FILL-IN is fixed parameter tractable (FPT) for the parameter k by giving an algorithm which runs in $\mathcal{O}(k^6 16^k + k^2 mn)$ time [16]. Following this, faster FPT algorithms were devised for the problem, with running times that have smaller constants in the base of the exponent [6, 7]. Very recently, the first and third authors of this paper developed a subexponential FPT algorithm for the problem which runs in $\mathcal{O}(2^{\mathcal{O}(\sqrt{k} \log k)} + k^2 nm)$ time [12].

In this paper we study kernelization algorithms for MINIMUM FILL-IN on different classes of sparse graphs. Kernelization can be regarded as systematic mathematical investigation of preprocessing heuristics within the framework of parameterized complexity. In parameterized complexity each problem instance comes with a parameter k and the parameterized problem is said to admit a *polynomial kernel* if there is a polynomial time algorithm (the degree of polynomial is independent of k), called a *kernelization* algorithm, that reduces the input instance down to an instance with size bounded by a polynomial $p(k)$ in k , while preserving the answer. This reduced instance is called a $p(k)$ *kernel* for the problem. If $p(k) = \mathcal{O}(k)$, then we call it a *linear kernel*. For example, for the instance (G, k) of PLANAR MINIMUM FILL-IN, where G is a planar graph and k is the parameter, the pair (G', k') is a linear kernel if G' is planar, the size of G' , i.e., the number of edges and vertices, is $\mathcal{O}(k)$, and there is a fill-in of G with at most k fill edges if and only if there is a fill-in of G' with at most k' fill edges. Kernelization has been extensively studied, resulting in polynomial kernels for a variety of problems. In particular, it has been shown that many problems have polynomial and linear kernels on planar and other classes of sparse graphs [2, 5, 20].

There are several known polynomial kernels for the MINIMUM FILL-IN problem [16] on general (not sparse) graphs. The best known kernelization algorithm is due to Natanzon et al. [17], which for a given instance (G, k) outputs in time $\mathcal{O}(k^2 nm)$ an instance (G', k') such that $k' \leq k$, $|V(G')| \leq 2k^2 + 4k$, and (G, k) is a YES instance if and only if (G', k') is. Note that not every kernelization algorithm for fill-in in general graphs produces a sparse kernel, even if the input is a sparse graph. For example, the algorithm of Natanzon et al. [17],

while reducing the number of vertices in the input graph G , introduces new edges. Thus the resulting kernel G' can be very dense. In order to obtain kernels on classes of sparse graphs, we have to design new kernelization algorithms which preserve the sparsity of the kernel.

Our Results. We provide kernelization algorithms for three important and increasingly general classes of graphs. For planar graphs, we obtain an $\mathcal{O}(k)$ kernel, and for graphs excluding a fixed graph as a minor and graphs of bounded degeneracy, a kernel of size $\mathcal{O}(k^{3/2})$. Our reduction rules are easy to implement. Small kernels for sparse graphs can be used as an argument explaining the successful behavior of several heuristics for sparse matrix computations. As a byproduct of our results, we obtain an approximation algorithm that, for an input planar graph with minimum fill-in of size k , produces a fill-in of size $\mathcal{O}(k \log k)$, which is within factor $\mathcal{O}(\log k)$ of optimal. For H -minor-free graphs our kernelization yields an approximation with the ratio $\mathcal{O}(\sqrt{k} \log k)$.

2 Preliminaries

All graphs in this paper are finite and undirected. In general we follow the graph terminology of Diestel [9]. For a vertex v in graph G , $N_G(v)$ is the set of neighbours of v , and for two non-adjacent vertices u, v , $N_G(u, v) \equiv N_G(u) \cap N_G(v)$. We drop the subscript G where there is no scope for confusion. For $S \subseteq V(G)$, we use $N(S)$ for the set of neighbours in $V(G) \setminus S$ of the vertices in S , and $N[S] \equiv N(S) \cup S$. We also use $G[S]$ to denote the subgraph of G induced by S , and $G \setminus S$ to denote the subgraph $G[V \setminus S]$.

The operation of *contracting* an edge $\{u, v\}$ of a graph consists of replacing its endpoints u, v with a single vertex which is adjacent to all the former neighbours of u and v in G . A graph H is said to be a *contraction* of a graph G if H can be obtained from G by contracting zero or more edges of G . Graph H is a *minor* of G if H is a contraction of some subgraph of G . A family \mathcal{F} of graphs is said to be *H -minor free* if no graph in \mathcal{F} has H as a minor. For $d \in \mathbb{N}$, a graph G is said to be *d -degenerate* if every subgraph of G has a vertex of degree at most d . A family \mathcal{F} of graphs is said to be of *bounded degeneracy* if there is some fixed $d \in \mathbb{N}$ such that every graph in the family is d -degenerate. Note that all graph properties discussed in this paper (being chordal, planar, H -minor free, and d -degenerate) are hereditary, i.e., are closed under taking induced subgraphs.

Minimal Separators. Let u, v be two vertices in a graph G . A set S of vertices of G is said to be a *u, v -separator* of G if u and v are in different components in the graph $G \setminus S$. The set S is said to be a *minimal u, v -separator* if no proper subset of S is a u, v -separator of G . A set S of vertices of G is said to be a (*minimal*) *separator* of G if there exist two vertices u, v in G such that S is a (minimal) u, v -separator of G .

Let S be a separator of a graph G . A connected component C of $G \setminus S$ is said to be *associated* with S , and is said to be a *full component* if $N(C) = S$.

The following proposition is an exercise in [14].

► **Proposition 1.** *A set S of vertices of a graph G is a minimal u, v -separator if and only if u and v are in different full components of $G \setminus S$.*

A set S of vertices of a graph G is said to be a *clique separator* of G if S is a separator of G , and $G[S]$ is a clique.

Minimal and minimum fill-in. *Chordal* or *triangulated* graphs are graphs containing no induced cycles of length more than three. In other words, every cycle of length at least four in a chordal graph contains a chord. Let F be a set of edges which, when added to a graph

G , makes the resulting graph chordal. Then F is called a *fill-in* of G , and the edges in F are called *fill edges*. A fill-in F of G is said to be *minimal* if no proper subset of F is a fill-in of G , and F is a *minimum* fill-in if no fill-in of G contains fewer edges. Notice that every minimum fill-in is also minimal, and so to find a minimum fill-in it is sufficient to search the set of minimal fill-ins.

► **Proposition 2.** [6] *Let G be a graph, and let S be a minimal separator of G such that $G[S]$ is a complete graph minus one edge, and there is a vertex v in $V(G) \setminus S$ which is adjacent to every vertex in S . Then there exists a minimum fill-in of G which contains the single missing edge in $G[S]$ as a fill-edge.*

The following proposition is folklore; for a proof see, e.g., Bodlaender et al.'s recent article on faster FPT algorithms for the MINIMUM FILL-IN problem [6].

► **Proposition 3.** *Let $\langle v_1, v_2, v_3, v_4, \dots, v_t \rangle$ be a chordless cycle in a graph G , and let F be a minimal fill-in of G . If $\{v_1, v_3\} \notin F$, then $\{v_2, v\} \in F$ for some $v \in \{v_4, \dots, v_t\}$.*

► **Proposition 4.** [23] *Let S be a minimal separator of G , let G' be the graph obtained by completing S into a clique, and let $E_S = E(G') \setminus E(G)$. Let C_1, C_2, \dots, C_r be the connected components of $G \setminus S$. Then $E_S \cup F$ is a minimal fill-in of G if and only if $F = \bigcup_{i=1}^r F_i$, where F_i is the set of fill edges in a minimal fill-in of $G'[N[C_i]]$.*

Parameterized complexity. A parameterized problem Π is a subset of $\Gamma^* \times \mathbb{N}$ for some finite alphabet Γ . An instance of a parameterized problem is of the form (x, k) , where k is called the parameter. A central notion in parameterized complexity is *fixed parameter tractability (FPT)* which means, for a given instance (x, k) , solvability in time $f(k) \cdot p(|x|)$, where f is an arbitrary function of k and p is a polynomial in the input size. We refer to the book of Downey and Fellows [11] for further reading on Parameterized Complexity.

Kernelization. A *kernelization algorithm* for a parameterized problem $\Pi \subseteq \Gamma^* \times \mathbb{N}$ is an algorithm that given $(x, k) \in \Gamma^* \times \mathbb{N}$ outputs in time polynomial in $|x| + k$ a pair $(x', k') \in \Gamma^* \times \mathbb{N}$, called the *kernel* such that $(x, k) \in \Pi$ if and only if $(x', k') \in \Pi$ and $\max\{k', |x'|\} \leq g(k)$, and $k' \leq k$, where g is some computable function. The function g is referred to as the size of the kernel. If $g(k) = \mathcal{O}(k)$, then we say that Π admits a linear kernel.

The kernels in this paper are obtained by applying a sequence of polynomial time reduction rules. We use the following notational convention: for each reduction rule, (G, k) denotes the instance on which the rule is applied, and (G', k') denotes the resulting instance. We say that a rule is *safe* if (G', k') is a YES instance if and only if (G, k) is a YES instance. We show that each rule is safe. We also show—in most cases—that the resulting graph is in the same class as G .

The remaining part of the paper is organized as follows. Sections 3, 4, and 5 give kernel algorithms for planar, d -degenerate, and H -minor free graphs, respectively. All three kernels use Rule 2 in Section 3, and Rule 6 in Section 4 is used in Section 5 as well. The kernels obtained are then used in Section 6 to get approximations algorithms for planar and H -minor free graphs. We conclude and state some open problems in Section 7.

3 A Linear Kernel for Planar Graphs

In this section we show that the planar minimum fill-in problem has a linear kernel. The kernel is obtained by applying four reduction rules. Rules 1, 2, and 3 are applied exhaustively,

while Rule 4 is only applied if none of the other three can be applied. At the end of this process, the algorithm either solves the problem (giving either YES or NO as the answer), or it yields an equivalent instance (G', k') ; $k' \leq k$ where G is of size $\mathcal{O}(k)$.

► **Reduction Rule 1.** [24] Let S be a minimal clique separator in G and let C_1, \dots, C_t be the connected components of $G \setminus S$. We set G' to be the disjoint union of the graphs G_1, G_2, \dots, G_t , where G_i is isomorphic to $G[N[C_i]]$, $1 \leq i \leq t$, and set $k' \leftarrow k$.

By Proposition 4, we have the following lemma.

► **Lemma 1.** *Rule 1 is safe.*

Since each of the connected components of graph G' produced by Rule 1 is an induced subgraph of G , it follows that if G is planar or d -degenerate, then G' has the same property. Our next rule deletes vertices which are not part of any chordless cycle; as we show later (Theorem 3), a vertex v satisfies the conditions of the rule if and only if it is not part of any chordless cycle in the graph. This rule can be inferred from previous work due to Tarjan [24] and Berry et al. [3].

► **Reduction Rule 2.** For a vertex v of G , let C_1, C_2, \dots, C_t be the connected components of $G \setminus N[v]$. If for every $1 \leq i \leq t$, the vertex set $N(C_i)$ is a clique in G , then set $G' \leftarrow G \setminus \{v\}$, $k' \leftarrow k$.

► **Lemma 2.** *Rule 2 is safe.*

Proof. Let H be a chordal graph obtained by adding k edges to G . Chordality is a hereditary property, and thus the graph $H' = H \setminus \{v\}$ is chordal. But H' is a triangulation of $G' = G \setminus \{v\}$, and since it is obtained by adding at most k edges, we have that G' has a fill-in of size at most $k' \leq k$.

For the opposite direction, let H' be a *minimal* triangulation obtained from G' by adding the set of fill edges F' , where $|F'| \leq k'$. Then the graph H obtained by adding F' to G is chordal. Indeed, if H was not chordal, it would contain a chordless cycle A of length at least 4 passing through v . Let w be a vertex of A not adjacent to v and let C be the connected component of $G \setminus N[v]$ containing w . The set $S = N_G(C)$ is a clique minimal separator in G and thus by Proposition 4, we can conclude that in H every path from w to v should go through some vertex of S . Hence the set S contains at least two non-consecutive (in A) vertices a and b of A . But S is a clique in G , and thus is a clique in H . Hence, a and b form a chord in A , which is a contradiction. Therefore, H is chordal. ◀

In Reduction Rule 2, we only remove a vertex, and thus this rule does not change hereditary properties of graphs, like being H -minor free. We now state some useful properties of graphs on which the above reduction rules cannot be applied.

► **Lemma 3.** *A vertex v in a graph G does not satisfy the conditions of Reduction Rule 2 if and only if v is part of a chordless cycle in G .*

Proof. Let v be a vertex in G which does not satisfy the conditions of Reduction Rule 2. Then there exists a connected component C of $G \setminus N[v]$ such that $N(C)$ contains two non adjacent vertices, say $x, y \in N(v)$. Let P be a shortest path from x to y in $G[C \cup \{x, y\}]$. Since x and y are not adjacent, the path P is of length at least two; let $P = \langle x = v_1, v_2, \dots, v_\ell = y \rangle$. Since P is an induced path, $\langle v, x = v_1, v_2, \dots, v_\ell = y \rangle$ is a chordless cycle containing v .

Conversely, let $v = v_1, v_2, v_3, \dots, v_{r-2}, v_{r-1}, v_r = v$ be a chordless cycle in G containing v , and let C be the connected component of $G \setminus N[v]$ which contains v_3 and v_{r-2} . The vertex set $N(C)$ does not contain the edge $\{v_2, v_{r-1}\}$ and hence is not a clique. ◀

► **Lemma 4.** *Let G be a graph to which Rule 2 cannot be applied, and let F be an edge set such that $H = (V, E \cup F)$ is chordal. Then for every vertex v in G , there either exists an edge $\{v, x\} \in F$, or an edge $\{u, w\} \in F$, where $u, w \in N(v)$.*

Proof. By Lemma 3 it follows that every vertex v in G is part of at least one chordless cycle $\langle v = v_1, v_2, v_3, v_4, \dots, v_t \rangle$. By Proposition 3, there is either a fill edge $\{v, v_i\} \in F$ or an edge $\{v_2, v_t\} \in F$, for $i \in \{3, \dots, t-1\}$. ◀

► **Reduction Rule 3.** [6] Let (G, k) be an input instance of MINIMUM FILL-IN. If G has a minimal separator S such that adding exactly one edge to $G[S]$ turns it into a complete graph, and there exists a vertex v in $V(G) \setminus S$ such that all vertices of S are adjacent to v , then

1. Turn $G[S]$ into a complete graph by adding one edge,
2. Apply Rule 1 on the resulting minimal clique separator, and
3. Reduce k by one.

The correctness of this rule is evident from Proposition 2 and Lemma 1. We now show that the rule preserves the planarity of the graph. Observe that if the input graph G is planar, then $|S| \leq 4$.

► **Claim 1.** Reduction Rule 3 preserves the planarity of the graph.

Proof. Let G, S be as in the statement of the rule, and let G' be the graph obtained by applying the rule to G . Let $\{u, v\}$ be the missing edge in $G[S]$. By Proposition 1, there are at least two full components, say C_1, C_2 , associated with S in G . Notice that for each $i = 1, 2$, there is a uv -path in G with all internal vertices contained in C_i . This implies that each of the connected components of the output graph G' is a minor of planar graph G , and thus is planar. ◀

► **Reduction Rule 4.** Let (G, k) be an input instance of MINIMUM FILL-IN, where none of the Rules 1, 2, and 3 can be applied. If $|V(G)| > 6k - 4$ then return a trivial NO instance.

► **Lemma 5.** *Reduction Rule 4 is safe.*

Proof. Let (G, k) be a YES instance where $G = (V, E)$ is planar and none of the Rules 1, 2, and 3 can be applied. We now argue that $|V| \leq 6k - 4$.

Let F be an edge set such that $|F| \leq k$ and $H = (V, E \cup F)$ is chordal, and let V_F be the set of at most $2k$ vertices that are incident to the edges in F . We then have:

► **Claim 2.** Each vertex $v \in V \setminus V_F$ is adjacent to at least three vertices of V_F .

Proof. Since Rule 2 cannot be applied on vertex v it follows that $N[v] \subsetneq V$. Let C be a connected component of $G \setminus N[v]$ and let $S = N(C)$ be the minimal separator of G separating vertices of C from v . Rules 1 and 3 cannot be applied on S , so the graph $G[S]$ is missing at least two edges $\{x_1, y_1\}$ and $\{x_2, y_2\}$. By finding a shortest path P from x_j to y_j in $G[C \cup \{x_j, y_j\}]$ we can create a chordless cycle consisting of P and x_j, v, y_j for $j \in \{1, 2\}$. By Proposition 3 every fill-in of a chordless cycle either adds an edge incident to vertex v on the chordless cycle or adds a fill edge between its two unique neighbours. By definition there is no fill edge in F incident to v , and thus both $\{x_1, y_1\}$ and $\{x_2, y_2\}$ are contained in F . Two edges have to be incident to at least three vertices, and the claim follows. ◀

We construct a new graph $B = (V, E_B)$ whose edge set E_B is a subset of E , such that $\{u, w\} \in E_B$ if and only if $\{u, w\} \in E$, $u \in V_F$, and $w \notin V_F$. The graph B is planar since it is a subgraph of planar graph G , and is bipartite by construction with the two partite sets being $V_1 = V_F$ and $V_2 = V \setminus V_F$. As noted before, $|V_1| \leq 2k$; we now bound $|V_2|$. Let \mathcal{F} be the set of faces in any fixed planar embedding of B . Let $s = \sum_{f \in \mathcal{F}} (\text{number of edges on the face } f)$. Since B is bipartite, each face has at least four sides, and so $s \geq 4|\mathcal{F}|$. Since each edge of B lies on at most two faces in the embedding, it is counted at most twice in this process, and so $s \leq 2|E_B|$. Thus $4|\mathcal{F}| \leq 2|E_B|$. From this and the well-known Euler's formula for planar graphs applied to B (namely, $|V| - |E_B| + |\mathcal{F}| \geq 2$; observe that B may not be a connected graph) we get $|E_B| \leq 2|V| - 4 = 2(|V_1| + |V_2|) - 4$. By Claim 2 each vertex in V_2 has degree at least 3 in B , and so $|E_B| \geq 3|V_2|$. Combining these we get $|V_2| \leq 2|V_1| - 4 = 4k - 4$, and so $|V| = |V_1| + |V_2| \leq 6k - 4$. \blacktriangleleft

We now argue that all executions of the rules can be performed in polynomial time. By Proposition 4, a minimal clique separator is a clique separator in every minimal triangulation of the given graph. A minimal triangulation can be constructed in $\mathcal{O}(nm)$ time [22] and the minimal separators of the triangulation which are also cliques in G can be enumerated in $\mathcal{O}(nm)$ time [4]. As a consequence Rule 1 can be executed in polynomial time. For the remaining three rules it is not hard to see that we can check, find an instance, and execute the rule in polynomial time.

The rules are applied exhaustively in the order they are described. Rule 1 is globally applied at most $n - 1$ times, since all minimal clique separators we split on, even across connected components, are the so called “non-crossing” minimal separators in the initial graph, and a graph on n vertices has at most $n - 1$ pairwise non-crossing minimal separators [19]. Each time Rule 1 is applied, at most n connected components are created, and each of them contains at most n vertices. Thus, Rule 2 is applied at most $\mathcal{O}(n^3)$ times. Rule 3 is applied at most k times as one fill edges is added each time, and finally Rule 4 is applied only once. Thus we get

► **Theorem 6.** *MINIMUM FILL-IN has a planar kernel of size $\mathcal{O}(k)$ in planar graphs.*

4 An $\mathcal{O}(k^{3/2})$ kernel for d -degenerate graphs

We now describe two reduction rules for d -degenerate graphs. The second among these is in fact an algorithm which specifies how to apply Rule 2 and the first rule of this section in tandem. Given a problem instance (G, k) where G is a d -degenerate graph, the second rule outputs an equivalent instance (G', k') such that $k' \leq k$ and $|V(G')| = \mathcal{O}(k^{3/2})$. However, these rules *do not* guarantee that the resulting graph G' is d -degenerate. We will later show how to obtain an equivalent d -degenerate graph from G' while keeping the size bounded by $\mathcal{O}(k^{3/2})$.

The next reduction rule says that if two non-adjacent vertices in an d -degenerate graph G have many common neighbours, then the missing edge between the two vertices belongs to every small fill-in of G .

► **Reduction Rule 5.** Let (G, k) be an instance where G is d -degenerate. Let u, w be two non-adjacent vertices in G , and let $b = |N(u, w)|$. If $(b/2)(b - 1 - 2d) > k$, then set $G' \leftarrow (V(G), E(G) \cup \{\{u, w\}\})$, $k' \leftarrow k - 1$.

► **Lemma 7.** *Rule 5 is safe.*

Algorithm 1 Reduction Rule 6 for d -degenerate graphs.

```

1: procedure RULE6( $G, k$ )  $\triangleright G$  is assumed to be  $d$ -degenerate.
2:   while (Rule 2 applies to  $(G, k)$  and the vertex  $u$ ) do
3:      $G \leftarrow G \setminus \{u\}$ 
4:    $F'_0 = \emptyset$ 
5:   for (each nonadjacent pair  $x, y \in V(G)$ ) do
6:     if (Rule 5 applies to  $(G, k)$  and the non-adjacent vertices  $x, y$ ) then
7:        $F'_0 = F'_0 \cup \{\{x, y\}\}$ 
8:    $G' \leftarrow (V(G), E(G) \cup F'_0, k' \leftarrow k - |F'_0|)$ 
9:    $D_0 = \emptyset$ 
10:  while (Rule 2 applies to  $(G', k')$  and the vertex  $u$ ) do
11:     $G' \leftarrow G' \setminus \{u\}, D_0 = D_0 \cup \{u\}$ 
12:   $F_0 = E(G') \cap F'_0$ 
13:  if  $k' < 0$  or  $|V(G')| > 2k + k(2\sqrt{k} + 2d + 1)$  then
14:    return a trivial NO instance.
15:  else
16:    return  $(G', k')$ 

```

Proof. Let F be a fill-in of G of size at most k . We claim that $\{u, w\} \in F$. For, if $\{u, w\} \notin F$, then let H be the chordal graph obtained by adding the edges in F to the graph G . Since u and w are non-adjacent in H , there exists an u, w -separator in H , and every minimal u, w -separator in H contains all the vertices in $N(u, w)$. Since H is chordal, every minimal separator in H is a clique [10], and so the vertex set $N(u, w)$ induces a clique in H . Hence the subgraph $H[N(u, w)]$ contains $(b-1)b/2$ edges, where $b = |N(u, w)|$. Since G is d -degenerate, the subgraph $G[N(u, w)]$ contains at most db edges. Thus $|F| \geq (b-1)b/2 - db = (b/2)(b-1-2d) > k$, a contradiction, and so $\{u, w\} \in F$. It immediately follows that $F \setminus \{\{u, w\}\}$ is a fill-in of G' of size at most $k-1$.

Conversely, if G' has a fill-in F' of size at most $k-1$, then $F' \cup \{\{u, w\}\}$ is a fill-in of G of size at most k . \blacktriangleleft

► **Reduction Rule 6.** Let (G, k) be an instance where G is d -degenerate. Set (G', k') to be the instance output by Algorithm 1.

► **Lemma 8.** *Rule 6 is safe.*

Proof. By Rule 2 it is safe to delete vertex u in Line 3. Let $e_1, e_2, \dots, e_{|F'_0|}$ be the set of edges in F'_0 . By Rule 5 it is safe to add edge e_1 to G and decrement k . Let our induction hypothesis be that it is safe to add edges e_1, e_2, \dots, e_{i-1} to G and reduce k by $i-1$, and let us argue that it is also safe to add edges e_1, e_2, \dots, e_i and reduce k by i . Let $k_{i-1} = k - (i-1)$. Edge $e_i = \{x, y\}$ was added to F'_0 because $(b/2)(b-1-2d) > k$ where $b = |N_G(x, y)|$. In the extreme case, all the edges e_1, e_2, \dots, e_{i-1} are added between vertices in $N_G(x, y)$, but $(b/2)(b-1-2d) - (i-1) > k - (i-1) = k_{i-1}$ and thus it is safe to add edge e_i as well and reduce k_{i-1} by 1. We can now conclude that (G', k') in Line 8 is a YES instance if and only if (G, k) is. Finally by the safeness of Rule 2, instance (G', k') in Line 13 is a YES instance if and only if (G, k) is.

It remains to argue that we can safely return a trivial NO instance if $|V(G')| > 2k + k(2\sqrt{k} + 2d + 1)$, where G' is the graph in Line 13. Let us assume that (G', k') is a YES instance and let F be a set of edges such that $H = (V(G'), E(G') \cup F)$ is chordal and

$|F| \leq k'$. Let V_F be the set of vertices incident to edges of F , and let V_{F_0} be the set of vertices incident to edges of F_0 . Notice that $|V_F| + |V_{F_0}| \leq 2k$ as (G', k') is a YES instance. By Line 10 in Algorithm 1, Rule 2 is applied exhaustively, and thus by Lemma 4 every vertex of $V(G') \setminus (V_F \cup V_{F_0})$ is contained in $N_{G'}(x, y)$ for some edge $\{x, y\} \in F$. In particular, notice that $N_{G'}(x, y) \setminus (V_F \cup V_{F_0}) \subseteq N_G(x, y)$. Since G' is reduced with respect to Rule 5 (See Line 6 of Algorithm 1), $|N_G(x, y)| = b < 2\sqrt{k} + 2d + 1$. To see this we notice that a clique on b vertices contains $b(b-1)/2$ edges while $G[N_G(x, y)]$ contains at most db edges. Thus if $b \geq 2\sqrt{k} + 2d + 1$ then $b(b-1)/2 - db = b/2(b-1-2d) \geq ((2\sqrt{k} + 2d + 1)/2)(2\sqrt{k}) > k$ which is a contradiction to $\{x, y\} \notin F'_0$. Summing up we have that $|V(G')| \leq |V_F \cup V_{F_0}| + \sum_{\{x, y\} \in F} |N_G(x, y)| \leq 2k + k(2\sqrt{k} + 2d + 1)$. \blacktriangleleft

Observe that Rule 5 — which is applicable only when the input graph is d -degenerate — adds an edge to the graph. The graph resulting from applying Rule 6 — which adds the edge set F'_0 found by applying Rule 5 — may thus *not* be d -degenerate. The graph output by Rule 6 can be modified to become d -degenerate while preserving the bound on its size, and this gives an $\mathcal{O}(k^{3/2})$ kernel for MINIMUM FILL-IN in d -degenerate graphs:

► **Theorem 9.** MINIMUM FILL-IN has a d -degenerate kernel of size $\mathcal{O}(k^{3/2})$ in d -degenerate graphs.

Proof. Since a 1-degenerate graph is a forest, and every forest has a fill-in of size zero — since the forest is chordal — we can assume without loss of generality that $d \geq 2$. Let (G, k) be an instance of MINIMUM FILL-IN where G is a d -degenerate graph. The kernelization algorithm applies Reduction Rule 6 – Algorithm 1 – to (G, k) to obtain an equivalent instance (G', k') . If (G', k') is the trivial NO instance returned by Line 14, then it is d -degenerate and its size is a constant, and the kernelization algorithm returns (G', k') itself as the kernel.

Now let (G', k') be a non-trivial instance returned by Line 16. Observe that G' is obtained from G by (i) deleting some vertices – Line 3, – (ii) adding edges F'_0 – Line 6, – and (iii) deleting vertices D_0 – Line 11. Edge set F_0 is defined in Line 12 as the set of edges in F'_0 with both endpoints in $V(G')$.

The kernelization algorithm constructs a new graph G'' from G' by doing the following for each edge $\{u, v\} \in F_0$: remove $\{u, v\}$, add two new vertices a_{uv}, b_{uv} , and make both these vertices adjacent to both u and v . The algorithm returns (G'', k'') as the kernel, where $k'' = k' + |F_0|$. Let G_1 be the graph G' where edge set F_0 is removed.

To see that (G'', k'') satisfies all the requirements, note that G_1 is d -degenerate by the hereditary property of d -degenerate graphs, and $G' = (V(G'), E(G_1) \cup F_0)$. The graph G'' is d -degenerate since it can be obtained from G_1 by adding a sequence of vertices, each of degree two. Since each edge in F_0 corresponds to two new vertices in G'' , $|V(G'')| = |V(G_1)| + 2|F_0| \leq 4k + k(2\sqrt{k} + 2d + 1)$.

It remains to argue that (G', k') is a YES instance if and only if (G'', k'') is. If (G', k') is a YES instance, then let F' be a fill-in of G' of size at most k' , and let H' be the chordal graph obtained by adding the edges in F' to G' . Let $F'' = F_0 \cup F'$, and let H'' be the graph obtained by adding the edges in F'' to the graph G'' . Observe that H'' can be obtained from the chordal graph H' by adding a sequence of vertices of degree two each, each of which is adjacent to the two end-points of some edge in F_0 . It follows that H'' is chordal — any potential chordless cycle in H'' has to contain one of these new vertices, but every cycle passing through such a vertex has the respective edge in F_0 as a chord. Thus F'' is a fill-in of G'' of size at most $|F_0| + k' = k''$.

Conversely, let (G'', k'') be a YES instance. Observe that for each $\{u, v\} \in F_0$, the vertex set $S = \{u, v\}$ satisfies all the conditions of Proposition 2 in G'' — S is a minimal

a_{uv}, b_{uv} -separator (using the notation of the proof of Theorem 9), $G''[S]$ is missing the one edge which will make it a clique, and the vertex $a_{uv} \in V(G'') \setminus S$ is adjacent to every vertex in S . So there exists a *minimum* fill-in F'' of G'' such that $F_0 \subseteq F''$, and $|F''| \leq k''$. Let H'' be the chordal graph obtained by adding the edges in F'' to the graph G'' , and let H' be the graph obtained by deleting all the vertices $\{a_{uv}, b_{uv} \mid \{u, v\} \in F_0\}$ from H'' . Then H' can be obtained by adding the edges in $F' = F'' \setminus F_0$ to G' , and H' is chordal by the hereditary property of chordality. Thus F' is a fill-in of G' of size at most k' . ◀

5 An $\mathcal{O}(k^{3/2})$ kernel for H -minor free graphs

It is known [25] that every H -minor free graph is d -degenerate for $d \leq \alpha h \sqrt{\log h}$, where $h = |V(H)|$ and $\alpha > 0$ is a constant. As we have already shown in Section 4, the application of Rules 2, 5, and 6 on d -degenerate graphs results in an equivalent instance (G', k') where G' has $\mathcal{O}(k^{3/2})$ vertices. However, this G' is not necessarily H -minor free or d -degenerate. In Theorem 9, we show how to transform G' into a d -degenerate graph without significantly increasing its size. We employ a somewhat more involved transformation to convert G' to an H -minor free problem instance on $\mathcal{O}(k^{3/2})$ vertices:

▶ **Theorem 10.** [\star]¹ *Let H be a fixed graph. MINIMUM FILL-IN has an H -minor free kernel of size $\mathcal{O}(k^{3/2})$ in H -minor free graphs.*

6 Approximation

As a byproduct of our kernelization algorithms, we obtain improved approximation algorithms for the MINIMUM FILL-IN problem on planar and H -minor free graphs. We need the following result of Chung and Mumford [8].

▶ **Proposition 5.** [8] *Let H be a fixed graph, and let G be an n vertex graph that is H -minor free. Then there is a triangulation H_T of G such that $|E(H_T)| = \mathcal{O}(n \log n)$, and such a triangulation can be found in polynomial time.*

Together with our improved kernels, this result yields approximate solutions for MINIMUM FILL-IN, with ratio $\mathcal{O}(\sqrt{k} \log k)$ for H -minor-free graphs, and with ratio $\mathcal{O}(\log k)$ for planar graphs.

▶ **Theorem 11.** *Let k be the minimum size of a fill-in of a graph G . There is a polynomial time algorithm which computes a fill-in of G of size $\mathcal{O}(k \log k)$ if G is planar and of size $\mathcal{O}(k^{3/2} \log k)$ if G is H -minor free for some fixed graph H .*

Proof. Let G be a planar graph. For each $k \in \{1, 2, \dots, n^2\}$, in this order, we run the algorithm of Theorem 6 on (G, k) , and compute the maximum value k^* of the parameter k for which the algorithm gives us a NO answer. This guarantees that there is no fill-in of G of size k^* . We then run the same algorithm on the instance $(G, k^* + 1)$ to obtain a planar kernel G' on at most $6(k^* + 1)$ vertices. Using Proposition 5, we obtain a fill-in of G' with at most $c(k^* + 1) \log(k^* + 1)$ edges, for some constant c . By making use of standard backtracking, the solution for G' can be transformed into a fill-in of G with $\mathcal{O}(k \log k)$ fill edges.

The arguments when G is an H -minor free graph are almost identical to the planar case. The only difference is that we use Theorem 10 instead, which provides us with an H -minor free kernel of size $\mathcal{O}(k^{3/2})$. ◀

¹ Proofs of results labelled with a \star have been deferred to a longer version of the paper.

7 Conclusion and Open Questions

In this paper we obtained new algorithms for MINIMUM FILL-IN on several sparse classes of graphs. Specifically, we obtained a linear kernel for the problem on planar graphs and kernels of size $\mathcal{O}(k^{3/2})$ in H -minor free graphs and in graphs of bounded degeneracy. Using these kernels, we obtained approximation algorithms with ratios $\mathcal{O}(\log k)$ for planar graphs, and $\mathcal{O}(\sqrt{k} \log k)$ for H -minor free graphs. These results significantly improve known kernelization and approximation results for this problem. We note that for any $g \in \mathbb{N}$, the same set of reduction rules and essentially the same argument as for the planar case shows that MINIMUM FILL-IN has a kernel of size $\mathcal{O}(k)$ in graphs of genus at most g . We conclude with a number of open questions.

MINIMUM FILL-IN on general graphs is NP-complete [26]. However, it is a very old open question if the problem is NP-complete on planar graphs [8]. It turns out that MINIMUM FILL-IN is NP-complete on bipartite 2-degenerate graphs.

► **Theorem 12.** [★] *The MINIMUM FILL-IN problem is NP-complete on bipartite 2-degenerate graphs.*

The complexity of the problem on planar and on H -minor free graphs is still open. From the approximation perspective, we leave the possibility of obtaining an $o(\log k)$ -approximation on planar graphs as an open problem.

From the perspective of kernelization, it would be very interesting to find out if there is a linear kernel for MINIMUM FILL-IN on H -minor free graphs. We also were not able to find any evidence that the existence of an $\mathcal{O}(k/\log k)$ kernel on planar graphs would contradict any complexity assumption. Can it be that the problem has a sublinear kernel?

Acknowledgements.

We thank our anonymous reviewers for pointing out a way to reduce the constant factor in Reduction Rule 4 from 22 to 6 with a simpler proof of Lemma 5, and for many other comments which helped in improving the presentation. F. V. Fomin acknowledges the support of the European Research Council (ERC) via grant “Rigorous Theory of Preprocessing”, reference 267959.

References

- 1 A. Agrawal, P. N. Klein, and R. Ravi. Cutting down on fill using nested dissection: provably good elimination orderings. *Graph Theory and Sparse Matrix Computation*, 56:31–55, 1993.
- 2 J. Alber, M. R. Fellows, and R. Niedermeier. Polynomial-time data reduction for dominating set. *Journal of the ACM*, 51(3):363–384, 2004.
- 3 A. Berry, J. P. Bordat, P. Heggernes, G. Simonet, and Y. Villanger. A wide-range algorithm for minimal triangulation from an arbitrary ordering. *Journal of Algorithms*, 58(1):33–66, 2006.
- 4 J. R. S. Blair and B. W. Peyton. An introduction to chordal graphs and clique trees. In *Graph Theory and Sparse Matrix Computations*, pages 1–30. Springer, 1993. IMA Volumes in Mathematics and its Applications, Vol. 56.
- 5 H. Bodlaender, F. V. Fomin, D. Lokshtanov, E. Penninkx, S. Saurabh, and D. M. Thilikos. (Meta) Kernelization. In *FOCS 2009*, pages 629–638. IEEE, 2009.

- 6 H. Bodlaender, P. Heggernes, and Y. Villanger. Faster parameterized algorithms for minimum fill-in. *Algorithmica*, pages 1–22, 2010. Available online. DOI:10.1007/s00453-010-9421-1.
- 7 L. Cai. Fixed-parameter tractability of graph modification problems for hereditary properties. *Inf. Process. Lett.*, 58(4):171–176, 1996.
- 8 F. R. K. Chung and D. Mumford. Chordal completions of planar graphs. *J. Comb. Theory, Ser. B*, 62(1):96–106, 1994.
- 9 R. Diestel. *Graph Theory*. Springer-Verlag, Heidelberg, third edition, 2005.
- 10 G. A. Dirac. On rigid circuit graphs. *Abh. Math. Sem. Univ. Hamburg*, 25:71–76, 1961.
- 11 R. G. Downey and M. R. Fellows. *Parameterized complexity*. Springer-Verlag, New York, 1999.
- 12 F. V. Fomin and Y. Villanger. Subexponential parameterized algorithm for minimum fill-in. Accepted at the ACM-SIAM Symposium on Discrete Algorithms (SODA 2012)., 2012.
- 13 M. R. Garey and D. S. Johnson. *Computers and Intractability, A Guide to the Theory of NP-Completeness*. W.H. Freeman and Company, New York, 1979.
- 14 M. C. Golumbic. *Algorithmic Graph Theory and Perfect Graphs*. Academic Press, New York, 1980.
- 15 P. Heggernes. Minimal triangulations of graphs: A survey. *Discrete Mathematics*, 306(3):297–317, 2006.
- 16 H. Kaplan, R. Shamir, and R. E. Tarjan. Tractability of parameterized completion problems on chordal, strongly chordal, and proper interval graphs. *SIAM J. Comput.*, 28:1906–1922, May 1999.
- 17 A. Natanzon, R. Shamir, and R. Sharan. A polynomial approximation algorithm for the minimum fill-in problem. *SIAM J. Comput.*, 30:1067–1079, October 2000.
- 18 T. Ohtsuki, L. K. Cheung, and T. Fujisawa. Minimal triangulation of a graph and optimal pivoting ordering in a sparse matrix. *J. Math. Anal. Appl.*, 54:622–633, 1976.
- 19 A. Parra and P. Scheffler. Characterizations and algorithmic applications of chordal graph embeddings. *Discrete Applied Mathematics*, 79(1-3):171–188, 1997.
- 20 G. Philip, V. Raman, and S. Sikdar. Solving dominating set in larger classes of graphs: Fpt algorithms and polynomial kernels. In *ESA 2009*, volume 5757 of *Lecture Notes in Comput. Sci.*, pages 694–705. Springer, 2009.
- 21 D. J. Rose. A graph-theoretic study of the numerical solution of sparse positive definite systems of linear equations. In R. C. Read, editor, *Graph Theory and Computing*, pages 183–217. Academic Press, New York, 1972.
- 22 D. J. Rose, R. E. Tarjan, and G. S. Lueker. Algorithmic aspects of vertex elimination on graphs. *SIAM J. Comput.*, 5:266–283, 1976.
- 23 Kloks. T., D. Kratsch, and J. Spinrad. On treewidth and minimum fill-in of asteroidal triple-free graphs. *Theor. Comput. Sci.*, 175(2):309–335, 1997.
- 24 R. E. Tarjan. Decomposition by Clique Separators. *Discrete Mathematics*, 55:221–232, 1985.
- 25 A. Thomason. The extremal function for complete minors. *Journal of Combinatorial Theory, Series B*, 81(2):318 – 338, 2001.
- 26 M. Yannakakis. Computing the minimum fill-in is NP-complete. *SIAM J. Alg. Disc. Meth.*, 2:77–79, 1981.